

Remarks

This responds to the final Office action mailed January 19, 2007 [“the Action”]. Reconsideration of the application is respectfully requested in view of the following remarks. Claims 1-33 are pending in the application. No claims have been allowed. No claims are amended. Claims 1, 12, and 33 are independent.

Cited Art

U.S. Patent No. 6,083,276 to Davidson (“Davison”) is entitled “Creating and Configuring Component-Based Applications Using a Text-Based Descriptive Attribute Grammar.”

Amendments

Claims 1, 19, and 33 have been amended to respond to the Action’s rejection under 35 U.S.C. § 101. Claim 18 was amended to respond to the Action’s rejection under 35 U.S.C. § 112. These amendments were not made in response to the Action’s rejections under 35 U.S.C. § 102, and thus do not, in and of themselves, necessitate a new ground of rejection. Instead, Applicant’s arguments, below, are sufficient to show why the Action’s rejections under 35 U.S.C. § 102 are improper. No other amendments are made. No new matter has been added.

Claim Rejections - 35 U.S.C. § 101

Claims 1-33 have been rejected by the Action under 35 U.S.C. § 101 as “leading to a non-statutory subject matter.” [Action, at page 3, § 3.] In particular, the Action alleges that the claimed “data domain” or “data domains” of independent claims 1, 19, and 33 “do[] not reasonably convey the realization of a practical application result, which is required to be concrete, and necessarily useful and tangible.” [*Id.*] In doing so, the Action alleges that there exists “no reasonable teaching that this domain data amounts to any real world usefulness of any form,” and that the claimed subject matter “amounts to an abstract and necessarily computer-based *internal data*.” [*Id.*, at page 2, § 3.]

Applicants respectfully note that the independent claims have been amended to recite conditions on values used during testing. For example, claim 1 now recites “targeting testing of

the computer program to use only values for the data structure element that fall within the data domain” An example of such a condition can be found at page 8:

It may be desirable to target the testing or verification of software by selecting or limiting the domain of data values to be used in the testing or verification process. For example, in a software program that may use the Person data type 100 defined in FIG. 1 it may be desirable to limit the domain of data values of the Age field 111 to a range between 1-100 because human beings rarely live beyond 100 years of age and testing the software using Age values beyond 100 years may be meaningless.

[Application, at page 8, lines 10-15.] With these amendments, the claims make clear that produced data domains are utilized to control testing. Thus, the claims as amended realize a “practical application result,” namely that using the data domain to control testing.

The Action also rejects claims 19 and 33 for “recit[ing] a system without providing a tangible hardware support to realize the functionality of the configuration manager being recited.” Claim 19 has been amended to recite “a computer apparatus configured to perform the actions of a domain configuration manager.” The Application describes examples of the use of computer apparatuses starting at page 20:

Various types of general purpose or specialized computer apparatus may be used with or perform operations in accordance with the teachings described herein. Actions described herein can be achieved by computer-readable media comprising computer-executable instructions for performing such actions.

[Application, at page 20, line 29 to page 21, line 3.] Claim 33, likewise recites a “computer apparatus.” As such, claims 19 and 33 as amended recite tangible hardware support.

For at least these reasons, Applicants believe that claims 1, 19, and 33 recite statutory subject matter. Likewise, claims 2-18 and 20-32, which depend from claims 1 and 19 respectively, also recite statutory subject matter. Applicants request that the Examiner therefore withdraw the rejection under 35 U.S.C. § 101 to claims 1-33.

Claim Rejection – 35 U.S.C. § 112

The Action rejects claim 18 as being indefinite, and in particular for reciting “data domains of the parameter” with “insufficient antecedent basis” for the claim language. Claim 18 has since been amended to recite “data domains of parameters of the method.” As such, the

claim 18 now recites language with proper antecedent basis. Applicants request that the rejection of claim 18 under 35 U.S.C. § 112 be withdrawn.

Claim Rejections - 35 U.S.C. § 102(b)

The Action rejects claims 1-33 under 35 USC § 102(b) as being anticipated by Davidson. For a 102(b) rejection to be proper, the cited art must show each and every element as set forth in a claim. (See MPEP § 2131.01.) However, the cited art does not describe each and every element. Accordingly, applicants request that the rejection be withdrawn. Claims 1, 19, and 33 are independent.

Claim 1

Claim 1 recites, in part:

A computer implemented method for producing a data domain for a data structure element of a computer program, the method comprising:

...
receiving a reflection of the computer program;
producing the data domain based on the domain configuration information and the program reflection;

[Emphasis added.] For example, the Application describes examples data domains:

One aspect of testing and verification of software (regardless of the type of automation) that is particularly challenging is the ability of a testing tool to define a finite domain of data to be used in the testing. For instance, if a program accepts as one of its data inputs the basic data type of Integer that would mean that for a testing tool to exhaustively test the software it would have to conduct the test by applying virtually an infinite number of different integers. However, that is not only costly and time consuming but it also may be meaningless. Thus, a tester may look to the data members or variables of a program such as its various fields or parameters of the type Integer and manually limit the domain of each such data member to a selected set of values. For instance, if Age is a field of the type Integer then a meaningful domain for such a data member may be limited to integers ranging from 1-100. Such finitization is applicable to virtually any type of data. Thus, the process of testing can be vastly improved by specifying, or configuring the domain of the various data structure elements related to software programs.

[Application, at page 1, line 30 to page 2, line 11.] The application proceeds to provide additional examples starting at page 8:

In the case of methods the domain may be an enumeration of a set of tuples of its parameters. A set of tuples is a collection of values for a set of variables. For example, a one member tuple is represented as (a_1) , an ordered pair is a two member tuple represented as (a_1, a_2) , a three member tuple is represented as (a_1, a_2, a_3) and so on. FIG. 3B illustrates a data domain 325 for the method Run 320 (defined in FIG. 1), which receives the parameters of Age and Weight. Thus, the data domain for the Run method 122 is a set of two member tuples of various combinations of values of Age and Weight. For example, these tuples may be as follows:

$$(Age, Weight) = [(10,50), (12,55), (45, 167) \dots (20, 150)]$$

...

The data domain for a data type is a simple enumeration of the possible values of the data type. For example, in case of a basic type like Integer the domain may be a set of integer values for the Integer type to be used in the program. In this case, the domain may be represented as follows:

$$[1, 2, 3, 4, 5, \dots 100]$$

[*Id.* at page 8, line 29 to page 9, line 22; emphasis added.]

Davidson, however, does not “produce” a “data domain” for a “data structure element of a computer program” “based on the domain configuration information and the program reflection” as recited in claim 1 because Davidson is directed to mapping attributes to programmatic elements for the production of applications themselves. Davidson is directed toward “a method for creating and configuring a component-based application through [a] text-based descriptive attribute grammar.” [Davidson, at Abstract.] Davidson goes on to describe the end products of its processes in its Summary:

In accordance with a preferred embodiment of the present invention, a parse tree is created from an application description file. Thereafter, the parse tree is transformed into a plurality of components corresponding to instances of classes in an application framework. The components are then initialized and further processed to launch the component-based application.

[Davidson, at column 4, lines 51-58; emphasis added.]

Applicants note that Figures 4B and 4C were noted with particularity by the Action in rejecting the above-emphasized language of claim 1, with the Action alleging that “mapping corresponding descriptor or attribute for a method or class reads on data domain.” [Action, at page 4, § 7.] However, Applicants find no indication in Davidson that a descriptor or attribute, as described in Davidson describes a “data domain” as recited in claim 1, nor does the Action point to any such indication. In fact, Davison does not contain the word “domain” from which it could

be implied that the terms “descriptor” or “attribute,” as used in Davidson, could read on data domains.

Additionally, Applicants note that the process of Figures 4B and 4C (which illustrates a sub-process of the process of Figure 4B) is directed to “transforming 404 the parse tree 204 into uninitialized components 212.” [Davidson, at column 21, lines 66-67.] Hence, the end result of the process of Figures 4B and 4C is a set of uninitialized components, which are later “process[ed] . . . to launch the component-based application,” at the process of Figure 4D [Davidson, at column 22, lines 8-9.] These Figures, as well as the passages of Davidson describing them, demonstrate two features of Davidson that show it does not describe the above-quoted language of claim 1.

First, the processes of Figures 4B-4D are directed to the generation of *an application*, not data domains, as claim 1 recites. Secondly, as the application is being created by the processes of Figures 4B-4D, this means that *there is no extant computer program when these processes are performed*. Thus, as the processes of Figures 4B-4D work to create and launch a computer application, it would be impossible for a sub-part of that process to “receiv[e] a reflection of the computer program” and then “produc[e] the data domain based on the domain configuration information and the program reflection,” as recited in claim 1. This is because *there is no computer program* at the point in Davidson’s techniques at which the data domains are alleged to be produced.

For at least these reasons, the cited passages of Davidson cannot describe “producing the data domain based on the domain configuration information and the program reflection.” Furthermore, Applicants do not find further such disclosure elsewhere in Davidson. Applicants thus request that the rejection of claim 1 be withdrawn and that the claim be allowed.

Claim 19

Claim 19 recites:

A system for producing a data domain for a data structure element of a computer program, the system comprising:
a computer apparatus configured to perform the actions of a domain configuration manager for receiving domain configuration information corresponding to the data structure element and *using a reflection of the computer*

program to produce and output the data domain for the data structure element according to the domain configuration information.

[Emphasis added.] In its rejection of claim 19, the Action cites to the same passages of Davidson as it does in its rejection of claim 1. Thus, for at least the reasons discussed above with regard to claim 1, Davidson does not describe each and every element of claim 19. Applicants thus request that the rejection of claim 19 be withdrawn and that the claim be allowed.

Claim 33

Claim 33 recites:

A system for producing a data domain for a data structure element of a computer program, the system comprising:
a computer apparatus configured to perform the actions of a domain configuration manager for receiving domain configuration information corresponding to the data structure element and *using a reflection of the computer program to produce and output the data domain for the data structure element according to the domain configuration information.*

[Emphasis added.] In its rejection of claim 33, the Action refers directly to its rejection of claim 1. Thus, for at least the reasons discussed above with regard to claim 1, Davidson does not describe each and every element of claim 33. Applicants thus request that the rejection of claim 33 be withdrawn and that the claim be allowed.

Dependent Claims

The Action additionally rejects claims 2-18 and 20-32 35 USC § 102(b) as being anticipated by Davidson. Each of these claims, however, depends from either independent claim 1 or 19 and recites patentably distinct subject matter not described by the cited art. However, Applicants do not belabor the language of the individual claims in the interest of brevity but instead note that for at least the reasons given above with respect to the independent claims, the dependent claims are allowable. Applicants request that the rejections of claims 16 and 19 be withdrawn and that the claims be allowed.

Request for Interview

In view of the preceding amendments and remarks, Applicants believe the application to be allowable. If any issues remain, however, the Examiner is formally requested to contact the undersigned attorney at (503) 226-7391 prior to issuance of the next communication in order to arrange a telephonic interview. This request is being submitted under MPEP § 713.01, which indicates that an interview may be arranged in advance by a written request.

Conclusion

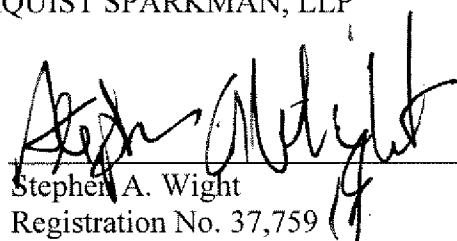
Claims 1-33 should be allowable. Such action is respectfully requested.

Respectfully submitted,

KLARQUIST SPARKMAN, LLP

One World Trade Center, Suite 1600
121 S.W. Salmon Street
Portland, Oregon 97204
Telephone: (503) 595-5300
Facsimile: (503) 595-5301

By



Stephen A. Wight
Registration No. 37,759